(12) UK Patent Application (19) GB (11) 2 363 044 (13) A

(43) Date of A Publication 05.12.2001

(21) Application No 0106700.8

(22) Date of Filing 16.03.2001

(30) Priority Data
(31) 09572140 (32) 16.05.2000 (33) US

(51) INT CL[7]
G06F 17/30

(52) UK CL (Edition S)
H4T TBLX

(56) Documents Cited
IBM Technical Disclosure Bulletin, Vol 39, No 8, August 1996 pages 57-60
IBM Technical Disclosure Bulletin, Vol 39, No 12, December 1996, pages 181-183

(58) Field of Search
UK CL (Edition S) H4T TBLA TBLX
Online: WPI, EPODOC, JAPIO, INSPEC, TDB, Internet

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Kevin B Haverlock
John E Lynde
Mark E Molander
Debapriya Ray

(74) Agent and/or Address for Service
R J Burt
IBM United Kingdom Limited, Intellectual Property Department, Hursley Park, WINCHESTER, Hampshire, SO21 2JN, United Kingdom

(54) Abstract Title
Method and computer system for selecting and displaying graphic objects

(57) To give the user of a computer system flexibility in generating displays of object trees, a user may initially select an object (see fig 5 not shown) appearing in a default tree display. A filter dialog box is summoned to allow the user to define custom object filters 98 or to select one of several predefined tree-level filters 96. The criteria associated with a selected filter are identified and retrieved and applied to all nodes (60a, 60b, 60c) on branches that are at levels subordinate to the level of the initially selected node (59). Once all of the relevant nodes (60a, 60b, 60c) are tested, a new tree display is generated which retains the tree structure but which shows only those nodes (64, 66, 68) which meet the selection criteria.

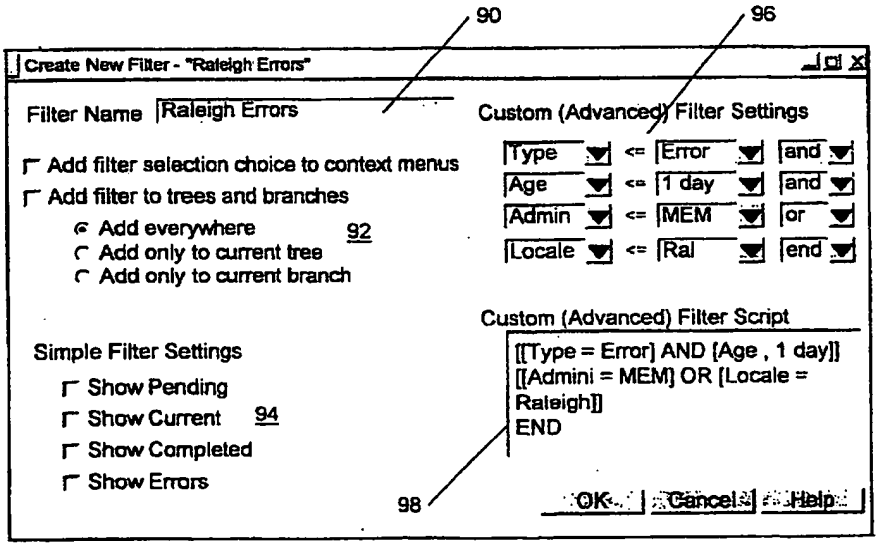

Fig. 9

GB 2 363 044 A

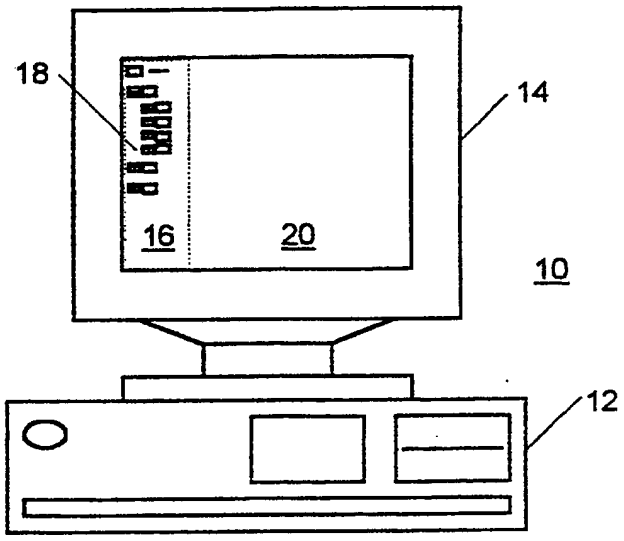
18
14
16
20
10
12

Fig. 1

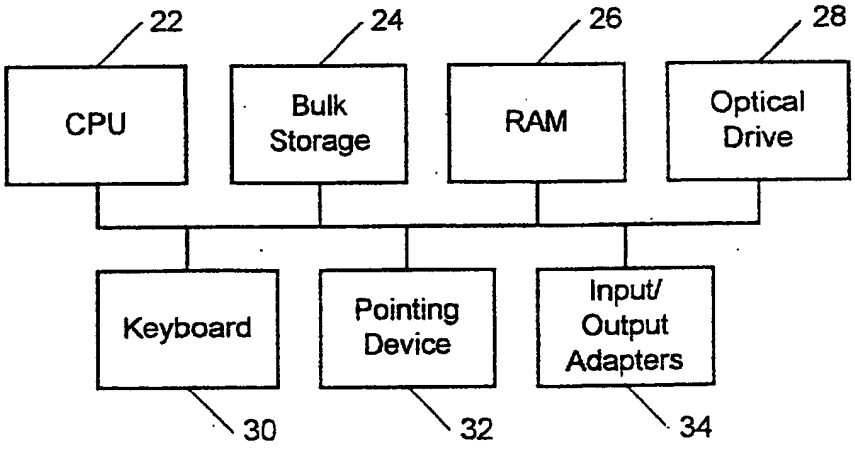
22
24
26
28
CPU
Bulk Storage
RAM
Optical Drive
Keyboard
Pointing Device
Input/ Output Adapters
30
32
34

Fig. 2

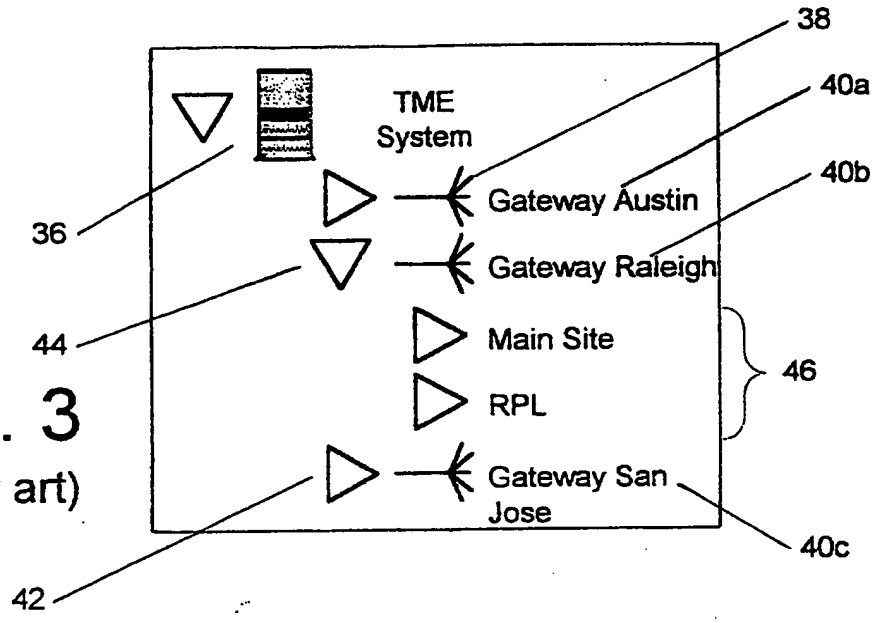

Fig. 3
(prior art)

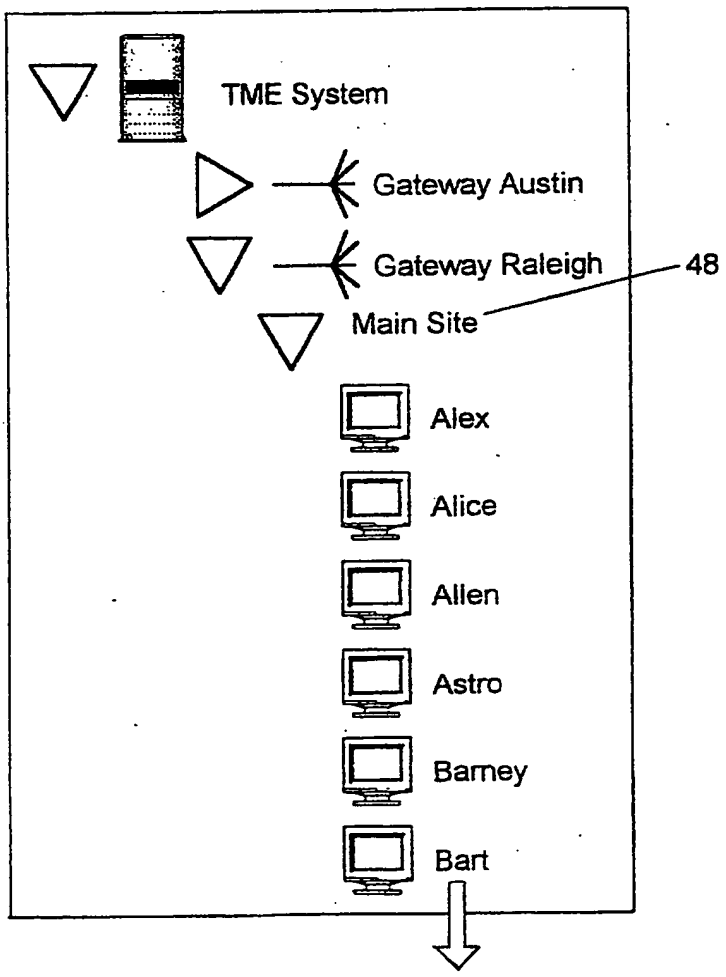

Fig. 4
(prior art)

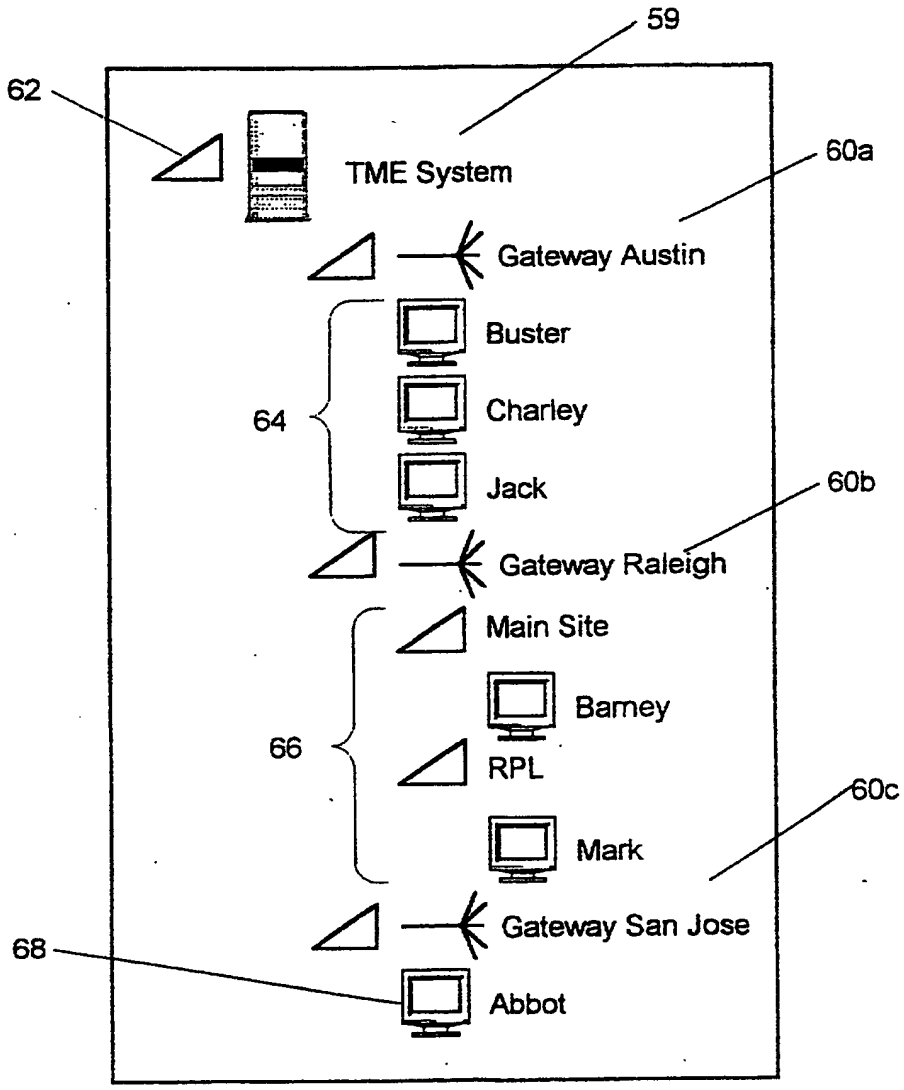
59
62
TME System
60a
Gateway Austin
Buster
Charley
64
Jack
60b
Gateway Raleigh
Main Site
Barney
66
RPL
60c
Mark
Gateway San Jose
68
Abbot

Fig. 5

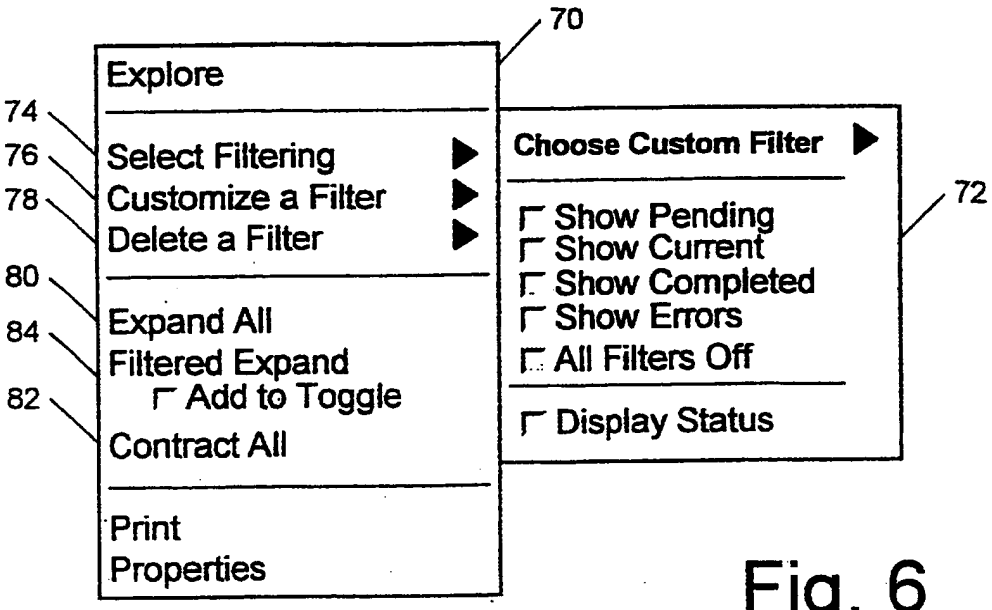
70
Explore
74
Select Filtering
76
Customize a Filter
78
Delete a Filter
80
Expand All
84
Filtered Expand
Add to Toggle
82
Contract All
Print
Properties
Choose Custom Filter
Show Pending
Show Current
Show Completed
Show Errors
All Filters Off
Display Status
72

Fig. 6

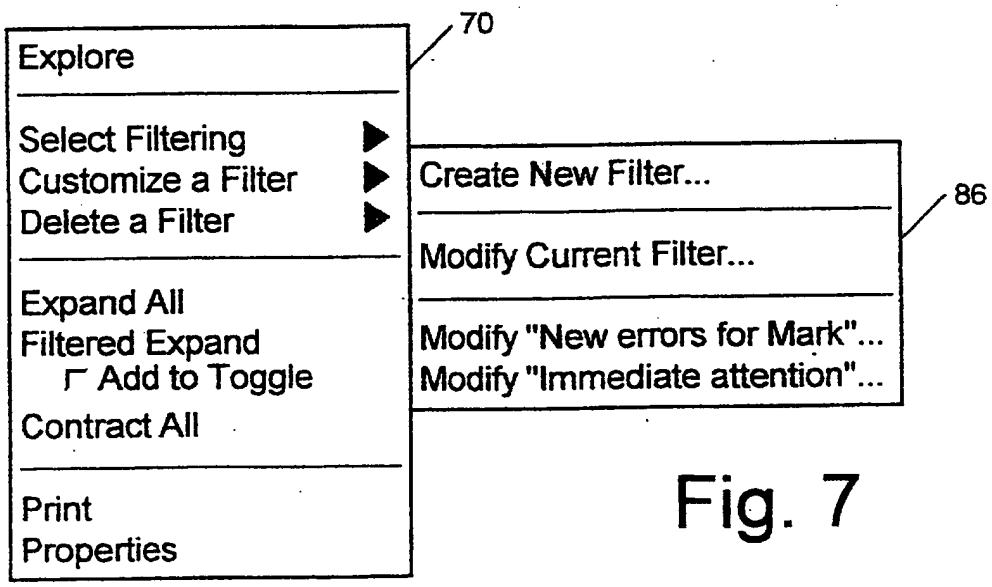
70
Explore
Select Filtering
Customize a Filter
Delete a Filter
Expand All
Filtered Expand
Add to Toggle
Contract All
Print
Properties
Create New Filter...
Modify Current Filter...
Modify "New errors for Mark"...
Modify "Immediate attention"...
86

Fig. 7

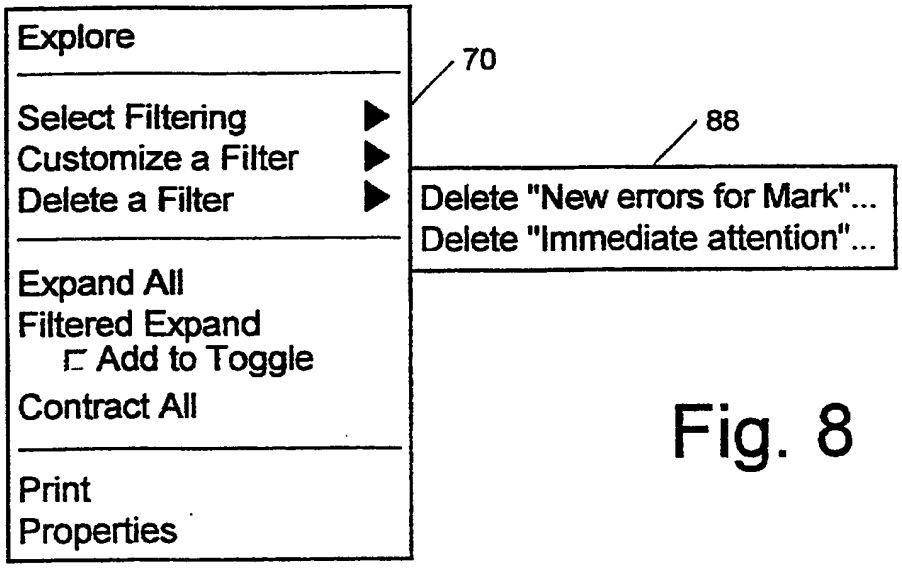
Explore
Select Filtering
Customize a Filter
Delete a Filter
Expand All
Filtered Expand
Add to Toggle
Contract All
Print
Properties
70
88
Delete "New errors for Mark"...
Delete "Immediate attention"...

Fig. 8

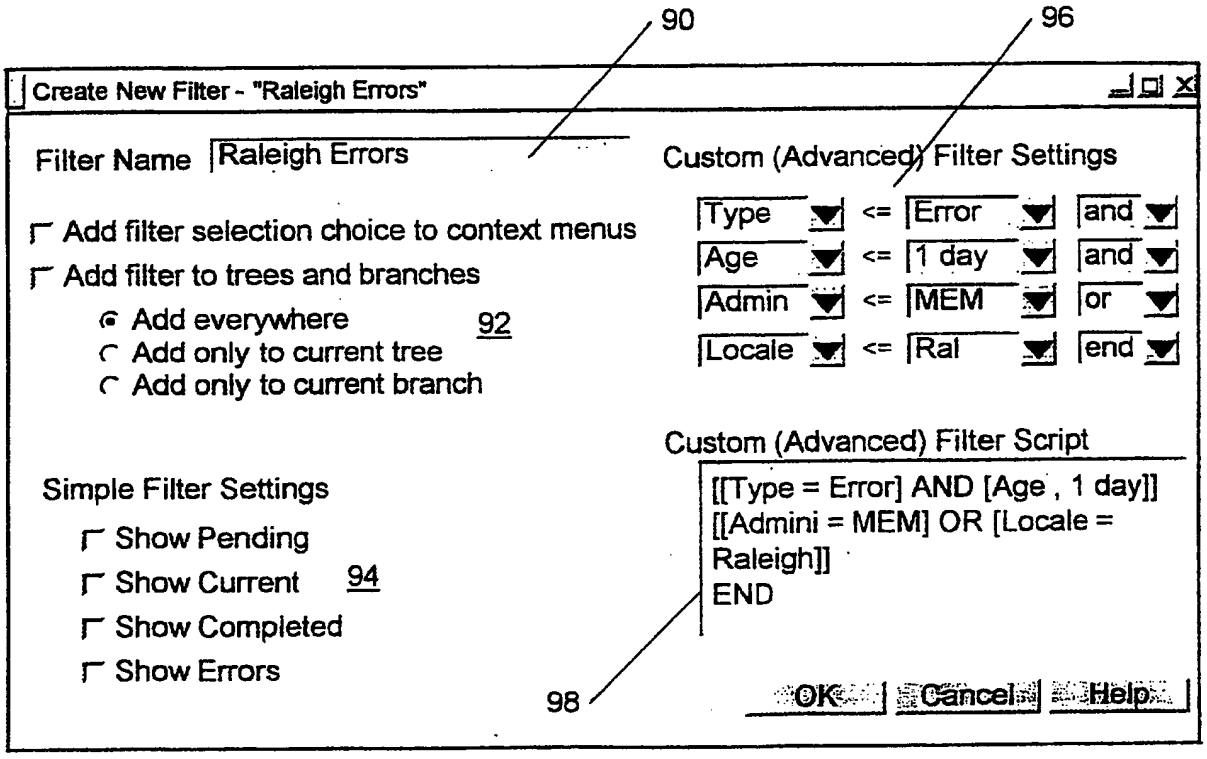
90
96
Create New Filter - "Raleigh Errors"
Filter Name
Raleigh Errors
Add filter selection choice to context menus
Add filter to trees and branches
Add everywhere
Add only to current tree
Add only to current branch
92
Simple Filter Settings
Show Pending
Show Current
Show Completed
Show Errors
94
Custom (Advanced) Filter Settings
Type <= Error and
Age <= 1 day and
Admin <= MEM or
Locale <= Ral end
Custom (Advanced) Filter Script
[[Type = Error] AND [Age , 1 day]]
[[Admini = MEM] OR [Locale = Raleigh]]
END
98
OK
Cancel
Help

Fig. 9

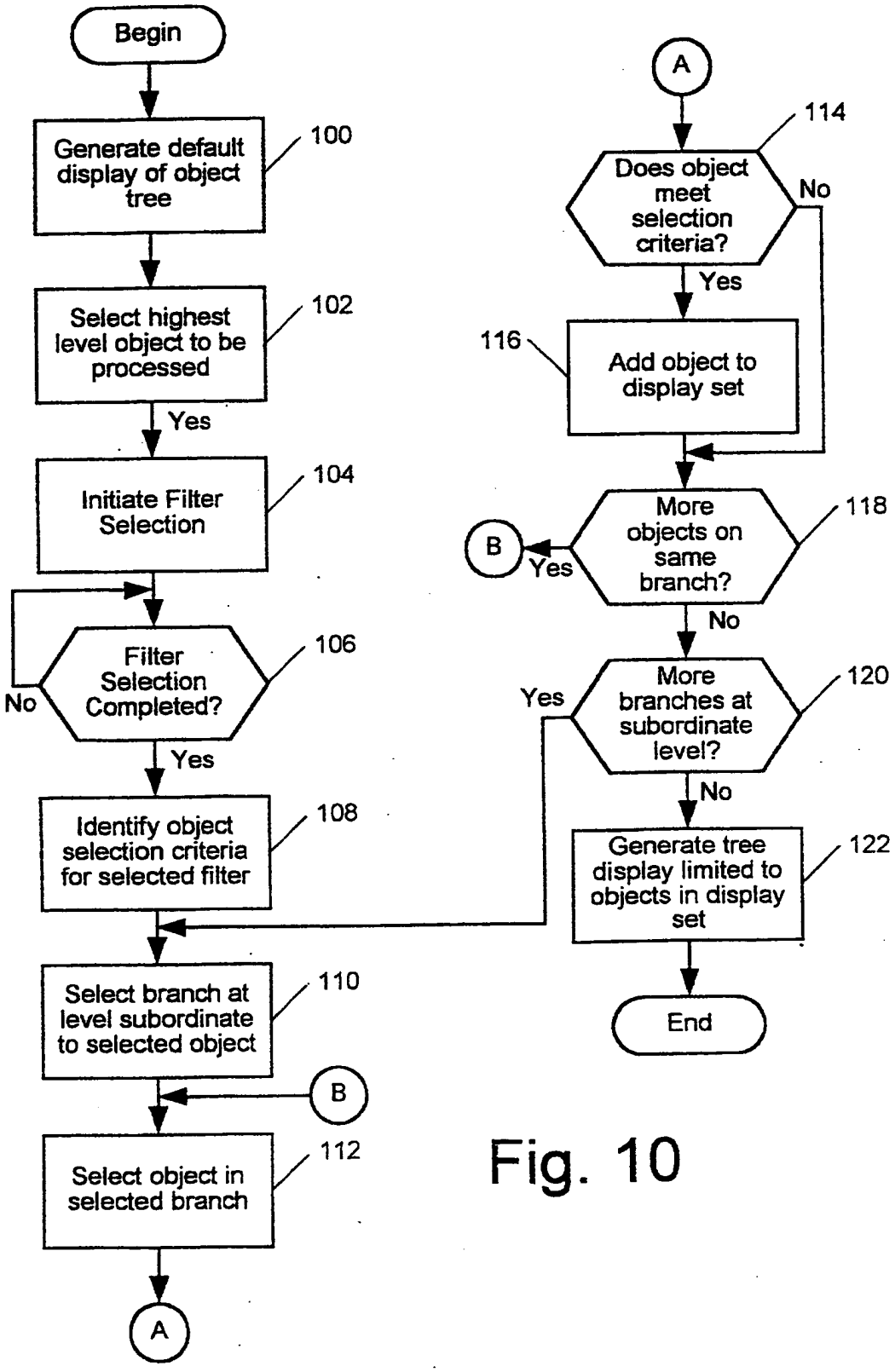
Begin
Generate default display of object tree
100
Select highest level object to be processed
102
Yes
Initiate Filter Selection
104
Filter Selection Completed?
106
No
Yes
Identify object selection criteria for selected filter
108
Select branch at level subordinate to selected object
110
B
Select object in selected branch
112
A
A
Does object meet selection criteria?
114
No
Yes
Add object to display set
116
More objects on same branch?
118
B
Yes
No
More branches at subordinate level?
120
Yes
No
Generate tree display limited to objects in display set
122
End

Fig. 10

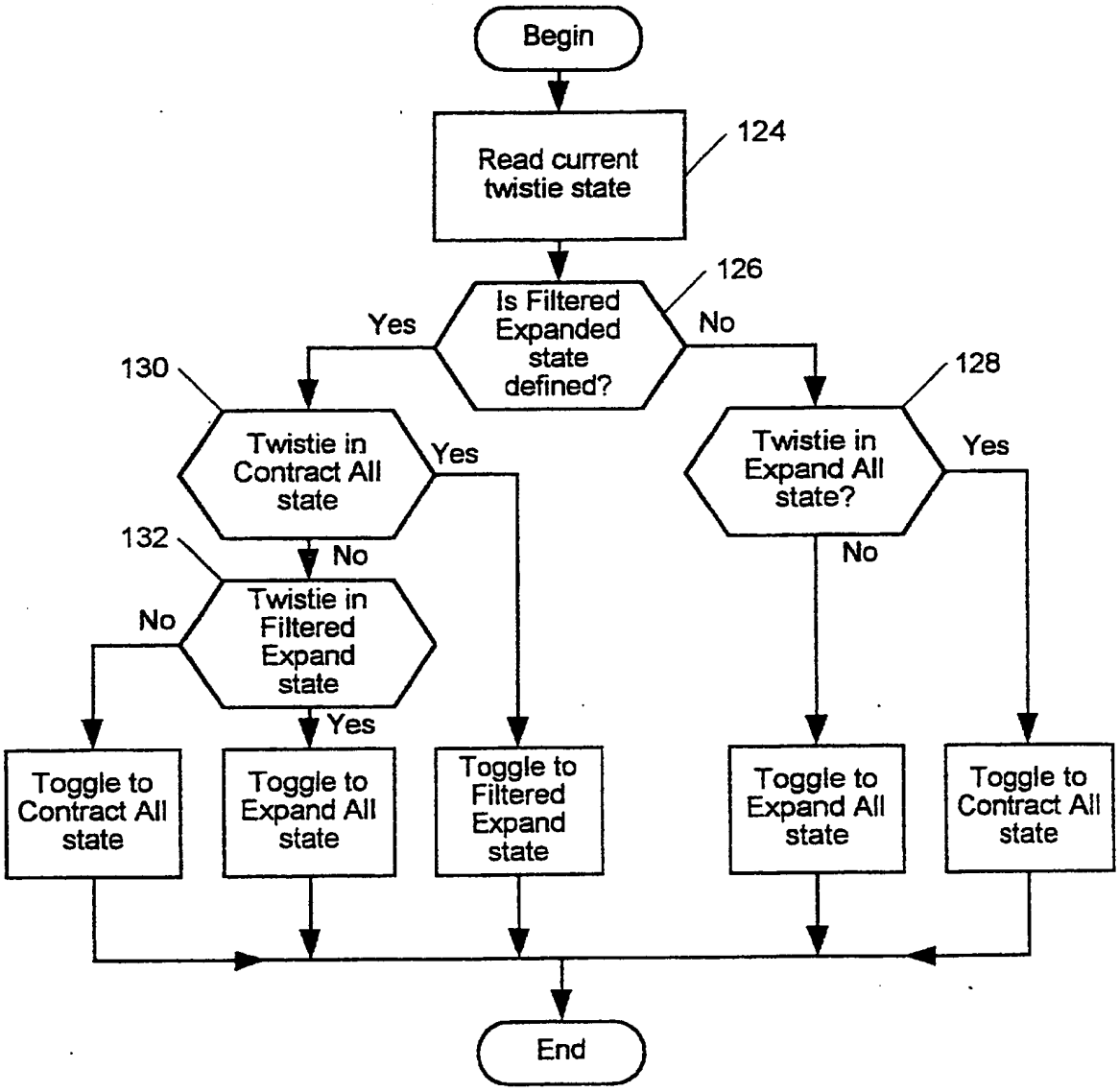
Begin
Read current twistie state
124
Is Filtered Expanded state defined?
126
Yes
No
130
Twistie in Contract All state
Yes
132
No
Twistie in Filtered Expand state
No
Yes
128
Twistie in Expand All state?
Yes
No
Toggle to Contract All state
Toggle to Expand All state
Toggle to Filtered Expand state
Toggle to Expand All state
Toggle to Contract All state
End

Fig. 11

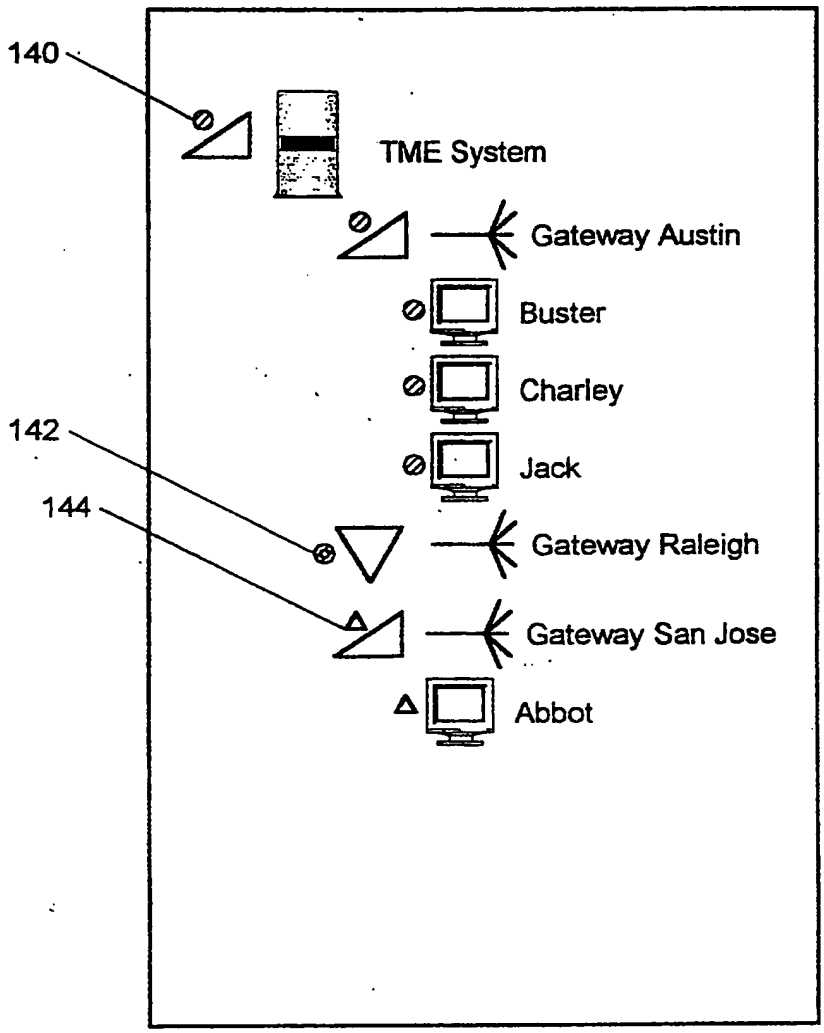
140
TME System
Gateway Austin
Buster
Charley
142
Jack
144
Gateway Raleigh
Gateway San Jose
Abbot

Fig. 12

METHOD AND COMPUTER SYSTEM FOR SELECTING AND DISPLAYING GRAPHIC OBJECTS

The present invention relates to a method and computer system for selecting and displaying graphic objects and more particularly to a computer system having a graphical user interface with in-line tree filtering.

In the field of data processing, resources are conventionally arranged in hierarchical form. A hierarchical arrangement is a multi-level arrangement in which a resource at a particular level is considered to contain or "own" one or more resources at the next lower level. The owning resource may, in turn, be owned or contained within a resource at the next higher level.

Perhaps the easiest way to understand a hierarchical system is by analogy to a filing system conventionally used to store paper documents. Most businesses include at least one filing room containing multiple filing cabinets. Each of those filing cabinets has several drawers. Each of those drawers is capable of holding several accordion folders. Each of those accordion folders is capable of holding a number of individual project folders. The filing room, filing cabinets, drawers, accordion folders and individual project folders form a multi-level hierarchy. Beginning at the lowest level, the individual project folders are contained in or "owned" by the accordion folder in which they are contained. All of the accordion folders in a drawer are considered to be "owned" by that drawer. All of the drawers in a particular filing cabinet are "owned" by that cabinet, etc.

In a data processing environment, the most ubiquitous example of a hierarchical system is the file system used for program or data files in a computer system. A computer system often contains multiple physical or logical storage drives. Each of those storage drives may be logically divided into a number of folders, sometimes referred to as directories. With one exception, every folder at every level in a hierarchical file system may contain only folders or folders plus one or more program or data files. The one exception is that the lowest level of a file system hierarchy contains, by definition, only program or data files.

Another example of a hierarchical data processing system is a distributed system. The highest level in a distributed system is a primary

network or extranet made up of the number of individual networks. Each of the individual networks may, in turn, contain smaller networks such as local area networks or LANs. Individual "end-user" devices such as workstations or printers may be connected directly to the LAN media. Alternatively, the individual "end user" devices may be connected to servers that are, in turn, connected to the LAN media.

In describing a hierarchical system in a data processing environment, it is conventional to refer to the hierarchy as a tree and to each object on the tree as a node. The highest-level object on the tree is referred to as the root node. Nodes other than the root node are described as being on a branch of the tree. Other objects on the tree are identified as parent nodes or child nodes depending upon their relationship to other nodes in the tree. A parent node is any node containing other lower-level nodes (child nodes). The same node may be both a child node (to a higher level parent node) and a parent node (to lower level nodes).

For anything other than the simplest of systems, the size of a tree can become quite large, making it difficult to find individual nodes. A user interface designer may choose to limit the presentation of tree information to a relatively small window at one edge of a display screen, which makes it even more difficult for the user to find nodes of interest in the tree. A user may have to scroll a tree both vertically and horizontally in an effort to locate a node of interest, a process which can be both cumbersome and frustrating since it can be difficult for the user to find the node to begin with and then to appreciate the logical relationship between a given node and the remainder of the tree, very little of which may be viewed at any given time.

A number of different tree representations have been proposed in an effort to make it easier for users to effectively deal with complex trees. Most of these representations require that an entire display screen be given over to the tree representation scheme, which may take the form of nested representations, quasi-three-dimensional representations or even hyperbolic representations of tree structure. Any tree representations which requires the use of an entire display screen is obviously of no practical value where a decision has already been made to limit the space available for tree representations to a relatively small window at one edge of a working area on the display screen. Where the space to be made available for a tree display is limited, a system user has had relatively few tools available for use in navigating through the tree structure.

One well-known tool for dealing with trees is the use of expand/collapse controls, also known as Atwistie@ controls, that may be visually represented by +/- icons or directional arrow icons. Selecting a twistie=s Aexpand@ icon adjacent a particular parent node causes the display to expand to include both the parent node and all direct child nodes. Selecting a twistie=s "collapse" icon adjacent a parent node in an already expanded display will cause all of the displayed child nodes to disappear, leaving only the original parent node.

There are at least two problems with expand/contract controls that limit their usefulness in navigating through a tree structure. First, the controls have to be selected one parent node at the time and then act to expand the display of only direct child nodes. Second, the controls operate on an all or nothing basis. When an expand control associated with a particular parent node is selected, every direct child node is shown on the display screen even if the system user is only interested in locating one or a few of the direct child nodes.

Efforts have been made to overcome some of the problems associated with twisties by using filters to limit the tree display to only nodes meeting the filter criteria. Known filter mechanisms come with their own sets of problems. Commonly, a filter is invoked using a dialog box that overlays part or all of the working area of the display screen. Also, known filters are applied only to the direct children of a particular parent node. These kinds of problems limit the flexibility and usefulness of known filter mechanisms.

According to a first aspect of the present invention there is provided a method for selecting and displaying one or more graphic objects in a set of graphic objects having predetermined locations in a hierarchical object tree, said method comprising the steps of: displaying at least a portion of the hierarchical object tree; monitoring user inputs identifying object selection criteria and a graphic object in the displayed portion; in response to the user inputs, limiting the display to the identified graphic object and all graphic objects which satisfy the identified object selection criteria and which occupy a subordinate position in the hierarchy of the identified graphic object.

Preferably, the object selection criteria is selected from a set of predefined selection criteria.

The user inputs may be monitored in response to a user's selection of a control associated with the identified graphic object. Preferably, the method includes the additional step of modifying the appearance of the control to indicate that only selected graphic objects are being displayed. The method may also include the additional step of adding an indicator adjacent a selected control, the added indicator having an appearance reflecting characteristics of the object selection criteria.

According to a second aspect of the present invention there is provided a computer system for enabling a user to select graphic objects within a hierarchical object tree for visual presentation, said system comprising: a processor; a memory coupled to the processor; a display coupled to the processor and having a display screen for visual presentation of graphic images; graphical user interface logic loaded into the memory for controlling the visual presentations on at least a portion of the display screen, said graphical user interface logic including: display logic for providing an initial visual presentation of a hierarchical object tree; and input logic for monitoring user inputs identifying object selection criteria and a graphic object included in the initial visual presentation, said display logic being responsive to the monitored user inputs to limit the visual presentation to the identified graphic object and all graphic objects which satisfy the object selection criteria and which occupy a subordinate position in the hierarchy of the identified graphic object.

Preferably, the computer system includes selection logic for enabling a user to predefine sets of selection criteria, each of said sets being identifiable by a monitored user input.

The display logic may provide a visual representation of a control associated with an identified graphic object and the input logic enables a user to select one of the predefined sets of selection criteria by repeatedly selecting the control. The display logic may further provide a visual indicator adjacent a selected control, said visual indicator reflecting the properties of object selection criteria associated with the control.

According to a third aspect of the present invention there is provided a computer program product stored on a computer readable storage medium, comprising computer readable program code means for performing the steps of: generating a display of at least a portion of a hierarchical

object tree wherein each object in the tree is represented by a graphic object having a predetermined location in the hierarchical object tree; monitoring user inputs identifying object selection criteria and a graphic object appearing in the generated display; in response to the user inputs, altering the display to include only the identified graphic object and all graphic objects which satisfy the identified object selection criteria and which occupy a subordinate position in the hierarchy of the identified graphic object.

Preferably, the computer readable program further enables a user to predefine sets of object selection criteria, each of said sets being selectable in response to a user input.

The computer readable program may further generate a graphic control object associated with the identified graphic object. The computer readable program may modify the appearance of the graphic control object to indicate that only selected graphic objects are being displayed.

The present invention is a filter tool that overcomes the problems of the known prior art. In one embodiment, the invention is implemented as a method of selecting and displaying one or more graphic objects in a set of graphic objects occupying predetermined locations in a hierarchical object tree. At least a portion of the hierarchical object tree is initially displayed. User inputs are monitored to determine when those inputs identify a graphic object in a displayed portion of the tree and a set of object selection criteria. When such user inputs are detected, the display is limited to the identified graphic object and all graphic objects which satisfy the identified object selection criteria and which are subordinate to the identified graphic object.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a representation of a workstation in which the present invention may be implemented;

Figure 2 is a simplified block diagram of essential components of the workstation of Figure 1;

Figure 3 is a representation of a conventional hierarchical object tree in a mostly collapsed state;

Figure 4 is a representation of the same object tree showing the appearance of the tree when a conventional expand operation is performed;

Figure 5 is a representation of an object tree following filtering in accordance with the present invention;

Figure 6 shows pulldown menus of the type used in selecting filters in accordance with the present invention;

Figure 7 shows pulldown menus of the type used in creating or modifying filters in accordance with the present invention;

Figure 8 shows pulldown menus of the type used in deleting existing filters in accordance with the present invention;

Figure 9 is a dialog box of the type that can be used to define custom filters in accordance with the present invention;

Figure 10 is a flowchart of steps performed in carrying out filtering operation in accordance with the present invention;

Figure 11 is a flow chart of the steps performed in extending the functionality of twistie controls to include the user-selected filters in accordance with the present invention; and

Figure 12 is a representation of an object tree showing filter status symbols that may be used to indicate the existence of a filter, types of filters and when set filters are being overridden by toggling of twistie controls in accordance with the present invention.

Referring to Figure 1, the present invention is a method for performing in-line filtering of object information visually presented on a workstation or console 10 including a system unit 12 and a monitor 14 having a display screen which can be logically divided into a window 16 in which an object tree 18 can be displayed and a working area 20 which normally displays video screens generated in the course of execution of one or more application programs.

The system unit 12 includes a number of conventional components, such as a central processor unit or CPU 22, bulk storage device 24 which conventionally takes the form of one or more hard drives located in the

system unit, random access memory 26 and an optical drive 28 implementing an optical storage technology such as CD-R, CD-RW or DVD. Depending on the technology implemented in drive 28, such a drive can be used to store an application to be executed and/or data resulting from the execution of an application. The system unit would typically further include input devices such as a keyboard 30 and a pointing device 32, most commonly a computer mouse. Finally, the system unit would include output adapters 34 to drive devices such as the monitor 14 shown in Figure 1 or a printer (not shown). The system unit could, of course, include many other types of adapters. The components shown in Figure 2 are intended only to represent the most basic components that would normally be found in virtually any system unit.

Figure 3 is a more detailed view of a small part of a known type of object tree. In the illustrated tree, the root or highest node in the tree is node 36. The root node 36 may also be referred to as a parent node as it has three child nodes 40a, 40b and 40c, all of which are identified as network gateway devices by icons such as icon 38 and by text further describing the devices.

Twistie controls are associated with the nodes on the object tree. A collapsed state is represented by a right facing triangle, such as triangle 42. Where a particular node is in the collapsed state, the tree shows only that node, but not any subordinate nodes. An expanded state is represented by a downward facing triangle, such as triangle 44. In the expanded state, the highest-level node in the branch is shown along with each direct child node in that branch. For example, the expanded state for node 40b shows node 40b itself and also subordinate nodes 46 (Main Site and RPL) defined as included within node 40b. While gateway nodes 40a and 40c may include as many or more subordinate nodes, the subordinates nodes do not appear in the object tree display since their parent nodes are defined as being in the collapsed state rather than the expanded state.

The tree representation shown in Figure 3 is one that will be familiar to users of some e-mail programs and network management programs. A perhaps more widely known but otherwise comparable form of tree representation is the one used in Windows operating system products available from Microsoft Corporation. In such systems, the expand/collapse controls are represented by +/- buttons, respectively. Depending upon how the system is configured, the individual nodes may be represented primarily by icons usually accompanied by descriptive text labels.

In most networks, the number of higher-level nodes, such as gateway nodes is limited. However, the number of lower-level nodes, including end nodes, may be considerably greater. Conventionally, when a twistie control for a particular node is expanded, that causes the system to attempt to display every node included within the next lower level. The number of nodes may be so great that it is impossible to show them all in the tree display window making navigation operations, such as scrolling, difficult.

As an example, referring to Figure 4, if the branch node 48 (Main Site) is expanded, the next lower or subordinate level may include so many nodes that not all of them can be shown concurrently in the tree window. This can be frustrating to a system administrator, particularly if the system administrator is not interested in all of the subordinate nodes but is instead interested only in nodes meeting certain criteria such as those that have completed assigned tasks or those that have encountered error conditions. Even if the icons used to represent the nodes are capable of taking on appearances indicative of operation in a particular state, the large numbers of nodes which are shown make it difficult for the system administrator to readily pick out a limited number of nodes meeting specified criteria.

In the described embodiment a filter tool makes it much easier for a system user, such as an administrator, to limit the tree display to only those nodes of current interest to him or her. The filter tool includes several pulldown menus and at least one dialog box that are used for selecting, creating, modifying and deleting filters. The structure and uses of these pulldown menus and dialog boxes will be described below following a description of the results of performing tree filtering operations using them.

The object filtering operations that are performed in response to selection of a branch- level or tree-level filter differs from conventional object filtering operations in several significant respects. For one thing, the filtering operation is applied to all nodes subordinate to a selected parent node and not just those nodes at the next lower level in the hierarchy. For another thing, only those nodes that meet the filter criteria will be displayed in the tree resulting from the filtering operations. For still another thing, the filter controls are inline controls directly displayed and available from within the tree itself. Finally, to make sure that the filtering operation is understood to be something other than the global expand or global collapse type of operation

known in the prior art, a unique twistie is used as part of the tree representation.

These characteristics are illustrated in Figure 5, which is a representation of the object tree which results when a selective filter is applied to the highest level root node and thus to every node at every subordinate level in the entire system. Assume, for purposes of illustration, that the system user has selected a tree-level filter that searches for nodes that have encountered some sort of error condition while attempting to complete assigned tasks.

When such a filter is applied to root node 59, its direct child nodes 60a, 60b and 60c are selectively expanded automatically as are all of the nodes subordinate to those nodes. However, the only objects that are displayed are those objects that meet the filter criteria. Thus, the expansion of gateway 60a results in the visual representation of only the few workstations shown in set 64. When gateway 60b is expanded, the expansion extends to directly attached nodes and all nodes subordinate to the directly attached nodes 66. Thus, the tree shows a single user at the Main Site and a single user at the RPL facility having workstations that have encountered error conditions. Similarly, the expansion of the gateway 60c will show a single workstation 68 that has encountered an error condition.

To make sure that the system user understands that the tree representation is not a representation of the complete (globally expanded) tree but rather of a limited tree resulting from a selective filter, a new control element 62, a right angle triangle, is used to indicate filtered expansion rather than global expansion. The new control, which might be referred to as a half- twistie in keeping with existing naming conventions, is applied to every node subordinate to the highest-level node in the selective filtering process. Of course, other types of visual representations for the twistie control are also possible.

Such a twistie control could cycle through more than two states. For instance, for the glyphs described, the twistie could cycle through three states; fully collapsed, filtered expanded, and fully expanded. These three states could also correspondingly be thought of as Aview all@, view some@ and Aview none@ in the tree. Each click on the currently active twistie control would cycle the twistie to the next of the three states.

Figure 6 illustrates a primary filter pulldown menu 70 that is selected, probably from an application toolbar, whenever tree-filtering operations are to be performed along with a cascaded menu 72 that is selected to perform specific filter selection operations. Several filter maintenance steps can be performed directly from the primary menu 72 including a Select step 74, a Customise step 76 and a Delete step 78, all of which are performed through secondary menus such as menu 72. Additionally, the primary menu includes controls that can be used to invoke specific tree-level or branch-level operations. The fields include such standard controls as an Expand All control 80 which, when selected, causes every node in the tree or branch to be displayed and an Contract All control 82 which, when selected, removes all objects from the tree or branch display other than the root object or objects, if a multi-rooted tree. In accordance with the invention, the primary menu also includes a Filtered Expand control 84 which will, when selected, cause the display tree to expand to include only objects that satisfy the selection criteria in the currently selected filter. The Filtered Expand operation can be invoked from menu 70. Alternatively, if the system user wants to be able to invoke a Filtered Expand operation using the tree=s twistie controls, control 84 includes a check box which allows the user to assign a currently selected filter to a third twistie representation; for example, the right triangle mentioned earlier. Once the current filter is assigned to third twistie representation, the system user can use the twisties to toggle among fully closed, filtered expanded and fully expanded object tree states.

Secondary menu 72 is selected whenever a user wants to select an existing filter, either a user-defined or custom filter or a standard predefined filter. The predefined filters will vary as function of the intended use of the filtering tool. The predefined filters shown in secondary menu 72 are typical of those that would be used in a network management application in which a user might want to track pending, current or completed tasks or tasks which encountered an error condition during their execution. Although not specifically shown, it can be assumed that selecting the control AChoose Custom Filter@ will cause a tertiary menu to be displayed listing available predefined filters.

Referring to Figure 7, where a user wants to customise a filter, a secondary menu 86 can be selected from primary menu 70. The secondary menu preferable gives the user the opportunity to create new filters, modify a

currently selected filter or to modify existing filters other than any currently selected filter.

Referring to Figure 8, if a user wants to delete existing filters, a secondary menu 88 can be called by selecting the Delete a Filter control on primary menu 70. The secondary menu 88 would preferably include a complete list of all existing filters that can be deleted under user control. Standard navigation techniques, such as vertical or horizontal scroll bars, might be used if the complete list is expected to occupy significant screen area.

It was mentioned earlier that a user could define a new filter. A dialog box such as the one illustrated in Figure 9 could be used for that task. The dialog box would preferably include a text box 90 into which the user would insert a name for the new filter, an array 92 of check boxes and radio buttons to be used for defining where the filter is to be effective, an array 94 of simple filter settings of the type already discussed with reference to the secondary menu 72, and an array 96 of drop-down lists that can be used to define a filter from predefined list entries and logical operators. Preferably, the array 96 is supplemented by a script box 98 that a user can employ to define any desired filter using a suitable scripting language.

Figure 10 is a flowchart of the steps that are performed in carrying out a selective filtering operation that might result in a tree display of the type just described with reference to Figure 5. Initially, a default display of the object tree is generated in a step 100. Typically, the default display will show a fully collapsed tree. When a user wishes to invoke a filter operation, the highest level tree object to which filtering is applied is selected in an operation 102 before filter selection is initiated in operation 104. The filter selection may include the step of selecting an existing filter, modifying an existing filter or defining a new filter. The exact sequence of steps that are performed in selecting a suitable filter are unnecessary to an understanding of the present invention. The completion of the filter selection operations is detected in an operation 106. When the desired filter has been set, the object selection criteria associated with the selected filter are identified by the system in step 108 and the actual filtering operation begins.

In an operation 110, a tree branch at a level subordinate to the level of the initially selected object is itself selected. An object in

that branch is identified in operation 112 and a test 114 is conducted to determine whether the selected object meets the previously identified selection criteria. If the object being considered does meet the selection criteria, the object is added to a display set in a step 116 and a test 118 determines whether there are more objects or nodes to be considered on the same branch. If test 114 indicates that the object does not meet the selection criteria, step 116 is bypassed. Until the test 118 indicates that every node in the selected branch has been tested, the program repeats a loop beginning at step 112 and ending at step 118

When every node in the selected branch has been tested, the program continues to a test 120 that determines whether there are more branches in the tree at levels subordinate to the level of the initially selected node. If there are, the program begins a second loop that begins at the input operation 110 and continues through operation 120 until every branch at a subordinate level has been selected. When all of the branches at levels subordinate to level of the initially selected node have been tested, a tree display is generated in a step 122 showing only those objects which satisfy the selection criteria along with labels associated with parent nodes for those objects.

As already noted, it may be desirable to allow the user to easily perform filtered expand operations by associating a predefined custom or standard filter to a Ahalf twistie@ symbol such as the right triangle discussed with reference to Figure 5. The flow chart shown in Figure 11 illustrates the system operations are performed when a user selects a twistie associated with a particular object in the tree. The current twistie state is read (operation 124) to determine whether the twistie currently represents a Contract All, Filtered Expand or Expand All state. The system is then tested (operation 126) to determine whether a Filtered Expand condition has been defined and whether the Filtered Expand is associated to a twistie representation.

If no Filtered Expand state is defined or if no association has been established between an existing filter and the twistie, then by definition the twistie can represent only Expand All or Contract All states. If a test 128 shows that the twistie is in an Expand All state already, the twistie is toggled to the next available state, namely, the Contract All state. If the twistie is not in an Expand All state, it must be in a Contract All state and would be toggled to the Expand All state.

If operation 126 had shown that a filter was associated to the twistie, then the twistie can represent one of three states, namely, the Contract All state, a Filtered Expand state and the Expand All state. Once operation 130 determines whether the twistie is in a Contract All state and operation 132 determines whether the twistie is in a Filtered Expand state, the next available twistie state is defined.

As noted earlier, filters can be applied at a branch level as well as a tree level and filters can be set and then overridden by toggling the twistie control associated with the tree or particular branches. To provide visual feedback to a user as to whether and where a filter is set, indicators of the types shown in Figure 12 may be included in the object tree display. A small filled circle 140 might be used to indicate that a filter has been applied to the objects in a tree or a particular branch in a tree. If a filter set for a particular branch has been overridden by toggling the twistie control associated with that branch, a hollowed circle or "donut" 142 might be used. Note that the twistie next to indicator 142 shows that the associated branch has been completely collapsed, which is something a user can accomplish by toggling the twistie control. The "donut" reminds that user of the existence of the original and overriding filters.

Since different filters can be applied to different branches in an object tree, it may be desirable to use different indicators to identify different filtered states. A red dot or donut of the type described might be appropriate for error-indicating filters. A small triangle 144 of a different colour, possibly yellow, might be appropriate where the filter is used to identify branches or objects operating in a warning state or condition.

While there has been described an embodiment of the invention, variations and modifications of the embodiment will occur to those skilled in the art. For example, while the process described above is initiated by first selecting the highest level object to which the filter is to be applied, one of ordinary skill in the art will recognise that it is well within the scope of the invention to trigger the process by first summoning a filter dialog box to select a desired filter and then to select a root node to which the selected filter will automatically be applied.

Improvements and modifications can be made to the foregoing without departing from the scope of the present invention.

CLAIMS

1. A method for selecting and displaying one or more graphic objects in a set of graphic objects having predetermined locations in a hierarchical object tree, said method comprising the steps of:

displaying at least a portion of the hierarchical object tree;

monitoring user inputs identifying object selection criteria and a graphic object in the displayed portion;

in response to the user inputs, limiting the display to the identified graphic object and all graphic objects which satisfy the identified object selection criteria and which occupy a subordinate position in the hierarchy of the identified graphic object.

2. A method as claimed in claim 1, wherein the object selection criteria is selected from a set of predefined selection criteria.

3. A method as claimed in claim 1 or 2, wherein the user inputs are monitored in response to a user's selection of a control associated with the identified graphic object.

4. A method as claimed in claim 3, including the additional step of modifying the appearance of the control to indicate that only selected graphic objects are being displayed.

5. A method as claimed in claim 4, including the additional step of adding an indicator adjacent a selected control, the added indicator having an appearance reflecting characteristics of the object selection criteria.

6. A computer system for enabling a user to select graphic objects within a hierarchical object tree for visual presentation, said system comprising:

a processor;

a memory coupled to the processor;

a display coupled to the processor and having a display screen for visual presentation of graphic images;

graphical user interface logic loaded into the memory for controlling the visual presentations on at least a portion of the display screen, said graphical user interface logic including

display logic for providing an initial visual presentation of a hierarchical object tree; and

input logic for monitoring user inputs identifying object selection criteria and a graphic object included in the initial visual presentation,

said display logic being responsive to the monitored user inputs to limit the visual presentation to the identified graphic object and all graphic objects which satisfy the object selection criteria and which occupy a subordinate position in the hierarchy of the identified graphic object.

7. A computer system as claimed in claim 6, further including selection logic for enabling a user to predefine sets of selection criteria, each of said sets being identifiable by a monitored user input.

8. A computer system as claimed in claim 7, wherein the display logic provides a visual representation of a control associated with an identified graphic object and the input logic enables a user to select one of the predefined sets of selection criteria by repeatedly selecting the control.

9. A computer system as claimed in claim 8, wherein the display logic further provides a visual indicator adjacent a selected control, said visual indicator reflecting the properties of object selection criteria associated with the control.

10. A computer program product stored on a computer readable storage medium, comprising computer readable program code means for performing the steps of:

generating a display of at least a portion of a hierarchical object tree wherein each object in the tree is represented by a graphic object having a predetermined location in the hierarchical object tree;

monitoring user inputs identifying object selection criteria and a graphic object appearing in the generated display;

in response to the user inputs, altering the display to include only the identified graphic object and all graphic objects which satisfy the identified object selection criteria and which occupy a subordinate position in the hierarchy of the identified graphic object.

11. A computer program product as claimed in claim 10, wherein the computer readable program further enables a user to predefine sets of object selection criteria, each of said sets being selectable in response to a user input.

12. A computer program product as claimed in claim 10 or claim 11, wherein the computer readable program further generates a graphic control object associated with the identified graphic object.

13. A computer program product as claimed in claim 12, wherein the computer readable program modifies the appearance of the graphic control object to indicate that only selected graphic objects are being displayed.

14. A method for selecting and displaying one or more graphic objects substantially as hereinbefore described with reference to the accompanying drawings.

15. A computer system for enabling a user to select graphic objects within a hierarchical object tree for visual presentation substantially as hereinbefore described with reference to the accompanying drawings.

16. A computer program product stored on a computer readable storage medium substantially as hereinbefore described with reference to the accompanying drawings.

INVESTOR IN PEOPLE

Application No: GB 0106700.8
Claims searched: All

Examiner: Joe McCann
Date of search: 26 September 2001

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.S): H4T(TBLA, TBLX)

Int Cl (Ed.7):

Other: Online: WPI, EPODOC, JAPIO, INSPEC, TDB, Internet

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	IBM Technical Disclosure Bulletin, Vol 39, No 8, August 1996, pages 57 to 60	
A	IBM Technical Disclosure Bulletin, Vol 39, No 12, December 1996, pages 181 to 183	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.